Rev 20050110a

CASRE Three-Page Product Documentation

Allen Nikora, Jet Propulsion Laboratory, March 16, 2005

What it is: CASRE is a tool for estimating and forecasting software reliability based on the failure history of a software system during test. It accepts two types of failure history as input: time between successive failures, or the number of failures in a test interval and the length of a test interval. CASRE implements about a dozen of the most widely-used software reliability models, and operates under all versions of Windows. CASRE is available at no charge from the Open Channel Foundation at http://www.openchannelfoundation.org/projects/CASRE_3.0.

Features: CASRE displays its results in both graphical and tabular form. In addition to its modeling capabilities, CASRE provides 1) trend tests (running arithmetic average and Laplace test) to determine whether it is appropriate to apply software reliability models to a software development effort, 2) capabilities for identifying the most appropriate model (goodness of fit, prequential likelihood ratio, model bias, and model bias trend), and 3) the ability to combine the results of several models according to a static or dynamic weighting scheme to increase model accuracy.

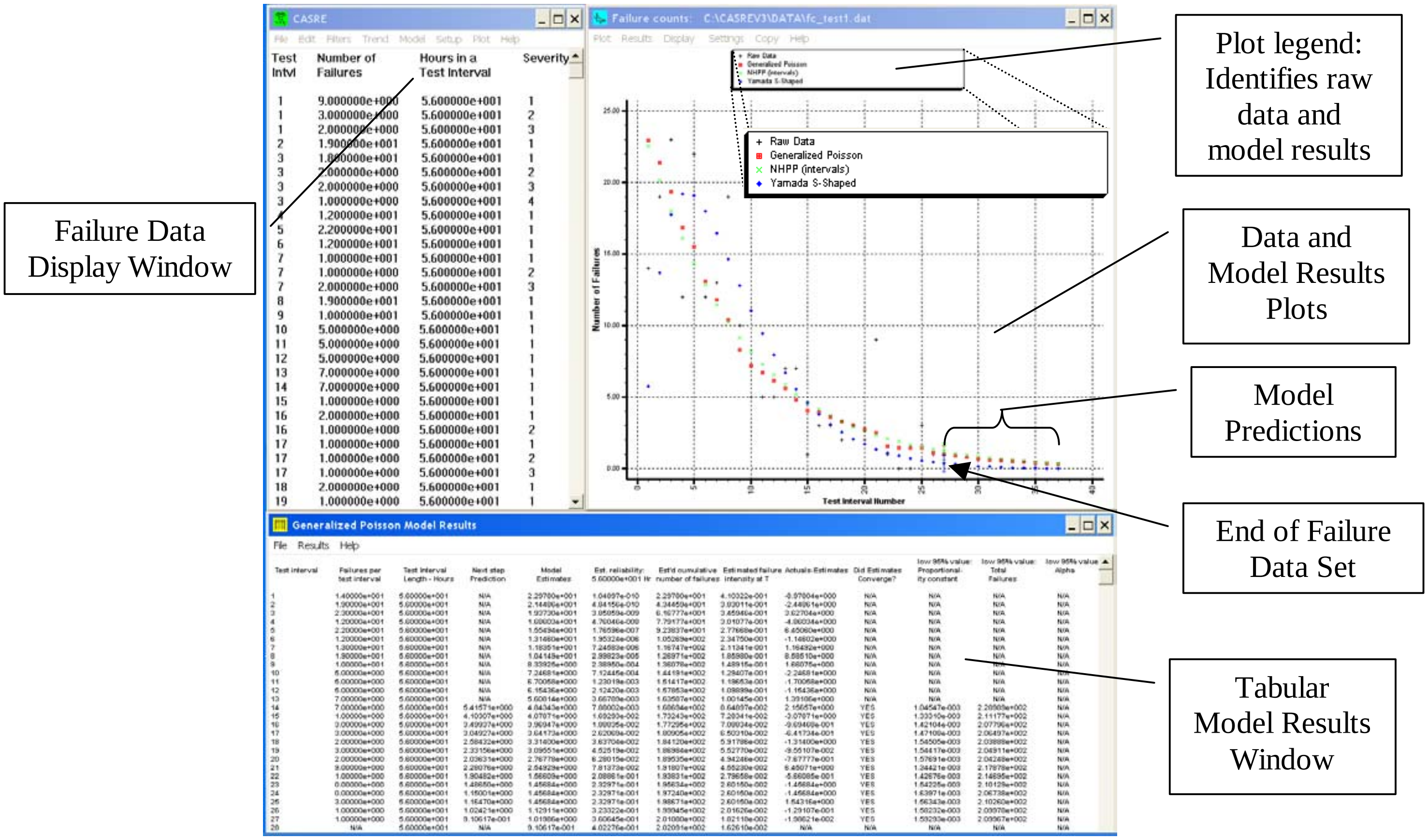
As shown in Figure 1, CASRE has three windows for displaying data, data analyses, and modeling results. The window in the upper left shows the failure history data on which the models implemented in CASRE will operate. CASRE's models can operate on two types of failure data – times between successive failures ("interfailure times") or the number of failures observed in a test interval of given length ("interval data"). The failure data shown in Figure 1 is interval data – the columns represent the test interval sequence number, the number of failures observed in that test interval, the length of the interval, and the severity of those failures, on an arbitrary scale from 1 to 9. If failures of different severities were observed during the same interval, there will be as many lines displayed for that interval as the number of different severities observed. Filters can be applied to view only those failures within a specified severity range, or having a single severity value. The filters preserve the correct interfailure time or failures per interval values. Software developers often want to know things such as the expected time to the next failure, the failure intensity, and reliability in terms of the most critical failures only – the filtering capability allows users to easily make these computations by simply applying the appropriate severity filter before applying software reliability models. Figure 1 displays the failure data, trend analysis results, and model results. In Figure 1, the failure data is plotted using the "+" symbol; the estimates and predictions of the models applied to that data are shown using colored symbols. In this case, CASRE makes predictions of failure intensity for the next five intervals beyond the end of the failure data; this is shown by the five colored points for each model beyond the I-beam symbol denoting the end of the failure data set. CASRE can display the model results in many different forms, including cumulative number of failures, failure intensity, interfailure times, and reliability. CASRE also displays quantitative model evaluation criteria such as goodness of fit, prequential likelihood ratio (how much more likely is it that one model will produce accurate predictions than another?), model bias, and model bias trend. CASRE also allows the user to rank models according to these criteria.

Finally, the bottom window shown in Figure 1 displays model results and raw data in tabular form. The model's estimates and predictions, parameter values, and evaluation criteria are displayed in form that can be exported to a comma separated value file, which in turn can be brought into a spreadsheet or statistical program for further analysis. This feature is aimed more at software reliability researchers wishing to investigate the behavior of models in detail, although software development organizations can also make use of results in this form.

Benefits: CASRE helps software developers determine when to stop testing by answering these questions: 1) Does the software currently meet its reliability requirements? 2) How much more effort is required to achieve the required reliability? 3) When will no more failures be observed?

Successes: The development and maintenance of CASRE was funded by the U. S. Air Force Operational Test and Evaluation Center (AFOTEC) for use in evaluating the reliability of several software systems developed for the Air Force. It has also been used by commercial software development organizations – Sun Microsystems, the Hewlett Packard printer division, Raytheon, and Motorola have all used CASRE to evaluate their software systems. Over 400 copies of CASRE have been downloaded from the Open Channel Foundation. The organizations that downloaded CASRE include SAIC, Alcatel, Siemens, Northrop Grumman, and Cisco Systems – however, it is not possible to determine how it is being used within those organizations.

Contexts in which it is best used: CASRE ***must*** be used after unit test, during the software integration or system integration phases. In addition, the testing effort ***must*** be similar to the way in which the system is expected to be operated during fielded use – if the input space is divided into equivalence classes, the frequency with which those classes are sampled during testing must be similar to the frequency with which those classes are expected to be sampled during operational use. Otherwise, the system may not exhibit reliability growth during test, and the models would not produce meaningful results. A software development effort using CASRE must also be diligent in recording failures observed during test as well as the elapsed test time.



Plot legend: Identifies raw data and model results

Failure Data Display Window

Data and Model Results Plots

Model Predictions

End of Failure Data Set

Tabular Model Results Window

Test Intvl	Number of Failures	Hours in a Test Interval	Severity
1	9.000000e+000	5.600000e+001	1
1	3.000000e+000	5.600000e+001	2
1	2.000000e+000	5.600000e+001	3
2	1.900000e+001	5.600000e+001	1
3	1.800000e+001	5.600000e+001	1
3	2.000000e+000	5.600000e+001	2
3	2.000000e+000	5.600000e+001	3
3	1.000000e+000	5.600000e+001	4
4	1.200000e+001	5.600000e+001	1
5	2.200000e+001	5.600000e+001	1
6	1.200000e+001	5.600000e+001	1
7	1.000000e+001	5.600000e+001	1
7	1.000000e+000	5.600000e+001	2
7	2.000000e+000	5.600000e+001	3
8	1.900000e+001	5.600000e+001	1
9	1.000000e+001	5.600000e+001	1
10	5.000000e+000	5.600000e+001	1
11	5.000000e+000	5.600000e+001	1
12	5.000000e+000	5.600000e+001	1
13	7.000000e+000	5.600000e+001	1
14	7.000000e+000	5.600000e+001	1
15	1.000000e+000	5.600000e+001	1
16	2.000000e+000	5.600000e+001	1
16	1.000000e+000	5.600000e+001	2
17	1.000000e+000	5.600000e+001	1
17	1.000000e+000	5.600000e+001	2
17	1.000000e+000	5.600000e+001	3
18	2.000000e+000	5.600000e+001	1
19	1.000000e+000	5.600000e+001	1

Test interval	Failures per test interval	Test interval Length - Hours	Next step Prediction	Model Estimates	Est. reliability: 5.60000e+001 Hr	Est'd cumulative number of failures	Estimated failure intensity at T	Actuals-Estimates	Did Estimates Converge?	low 95% value: Proportionality constant	low 95% value: Total Failures	low 95% value: Alpha
1	1.40000e+001	5.60000e+001	N/A	2.29700e+001	1.04097e-010	2.29700e+001	4.10322e-001	-8.97004e+000	N/A	N/A	N/A	N/A
2	1.90000e+001	5.60000e+001	N/A	2.14486e+001	4.04156e-010	4.34459e+001	3.93011e-001	-2.44861e+000	N/A	N/A	N/A	N/A
3	2.30000e+001	5.60000e+001	N/A	1.93730e+001	3.05059e-009	6.16777e+001	3.45946e-001	3.62704e+000	N/A	N/A	N/A	N/A
4	1.20000e+001	5.60000e+001	N/A	1.60603e+001	4.76046e-009	7.79177e+001	3.01077e-001	-4.06034e+000	N/A	N/A	N/A	N/A
5	2.20000e+001	5.60000e+001	N/A	1.55494e+001	1.76596e-007	9.23837e+001	2.77668e-001	6.45060e+000	N/A	N/A	N/A	N/A
6	1.20000e+001	5.60000e+001	N/A	1.31460e+001	1.95324e-006	1.05269e+002	2.34750e-001	-1.14602e+000	N/A	N/A	N/A	N/A
7	1.30000e+001	5.60000e+001	N/A	1.18351e+001	7.24683e-006	1.16747e+002	2.11341e-001	1.16492e+000	N/A	N/A	N/A	N/A
8	1.90000e+001	5.60000e+001	N/A	1.04149e+001	2.99823e-005	1.26971e+002	1.85980e-001	8.58510e+000	N/A	N/A	N/A	N/A
9	1.00000e+001	5.60000e+001	N/A	8.33926e+000	2.38950e-004	1.36078e+002	1.48916e-001	1.66075e+000	N/A	N/A	N/A	N/A
10	5.00000e+000	5.60000e+001	N/A	7.24681e+000	7.12445e-004	1.44191e+002	1.29407e-001	-2.24681e+000	N/A	N/A	N/A	N/A
11	5.00000e+000	5.60000e+001	N/A	6.70068e+000	1.23019e-003	1.51417e+002	1.19653e-001	-1.70068e+000	N/A	N/A	N/A	N/A
12	5.00000e+000	5.60000e+001	N/A	6.15436e+000	2.12420e-003	1.57853e+002	1.09899e-001	-1.15436e+000	N/A	N/A	N/A	N/A
13	7.00000e+000	5.60000e+001	N/A	5.60014e+000	3.66700e-003	1.63587e+002	1.00145e-001	1.39986e+000	N/A	N/A	N/A	N/A
14	7.00000e+000	5.60000e+001	5.41571e+000	4.84343e+000	7.86002e-003	1.68604e+002	8.64097e-002	2.15657e+000	YES	1.04547e-003	2.20509e+002	N/A
15	1.00000e+000	5.60000e+001	4.10307e+000	4.07071e+000	1.09293e-002	1.73243e+002	7.20341e-002	-3.07071e+000	YES	1.33310e-003	2.11177e+002	N/A
16	3.00000e+000	5.60000e+001	3.49937e+000	3.96947e+000	1.09035e-002	1.77295e+002	7.09034e-002	-9.69469e-001	YES	1.42104e-003	2.07796e+002	N/A
17	3.00000e+000	5.60000e+001	3.04927e+000	3.64173e+000	2.62069e-002	1.80905e+002	6.50310e-002	-6.41734e-001	YES	1.47109e-003	2.06497e+002	N/A
18	2.00000e+000	5.60000e+001	2.58432e+000	3.31400e+000	3.63704e-002	1.84120e+002	5.91786e-002	-1.31400e+000	YES	1.54505e-003	2.03888e+002	N/A
19	3.00000e+000	5.60000e+001	2.33156e+000	3.09551e+000	4.52519e-002	1.86984e+002	5.52770e-002	-9.55107e-002	YES	1.54417e-003	2.04911e+002	N/A
20	2.00000e+000	5.60000e+001	2.03631e+000	2.76778e+000	6.28015e-002	1.89535e+002	4.94246e-002	-7.67777e-001	YES	1.57691e-003	2.04248e+002	N/A
21	9.00000e+000	5.60000e+001	2.28076e+000	2.54929e+000	7.81373e-002	1.91807e+002	4.55230e-002	6.45071e+000	YES	1.24421e-003	2.17878e+002	N/A
22	1.00000e+000	5.60000e+001	1.90482e+000	1.56609e+000	2.08861e-001	1.93831e+002	2.79658e-002	-5.66085e-001	YES	1.42676e-003	2.14695e+002	N/A
23	0.00000e+000	5.60000e+001	1.48850e+000	1.45684e+000	2.32971e-001	1.95634e+002	2.60150e-002	-1.45684e+000	YES	1.54225e-003	2.10129e+002	N/A
24	0.00000e+000	5.60000e+001	1.16001e+000	1.45684e+000	2.32971e-001	1.97240e+002	2.60150e-002	-1.45684e+000	YES	1.63971e-003	2.06738e+002	N/A
25	3.00000e+000	5.60000e+001	1.16470e+000	1.45684e+000	2.32971e-001	1.98671e+002	2.60150e-002	1.54316e+000	YES	1.66343e-003	2.10260e+002	N/A
26	1.00000e+000	5.60000e+001	1.02421e+000	1.12911e+000	3.23322e-001	1.99945e+002	2.01626e-002	-1.29107e-001	YES	1.58232e-003	2.09978e+002	N/A
27	1.00000e+000	5.60000e+001	9.10617e-001	1.01986e+000	3.60645e-001	2.01080e+002	1.82110e-002	-1.98621e-002	YES	1.59293e-003	2.09967e+002	N/A
28	N/A	5.60000e+001	N/A	9.10617e-001	4.02276e-001	2.02091e+002	1.62610e-002	N/A	N/A	N/A	N/A	N/A

Figure 1 - CASRE On-Screen Appearance

Compare with alternative known products or technologies. There are several commercially and publicly available software reliability tools. These include SMERFS (Statistical Modeling and Estimation of Reliability Functions for Software), SoRel (developed by LAAS in France), and the set of tools developed by Ann Marie Neufelder. The best-known is SMERFS, developed by Bill Farr at the Naval Surface Warfare Center (http://www.slingcode.com/smerfs/). Indeed, CASRE uses the mathematical modeling libraries that were developed for SMERFS. CASRE differs from the current version of SMERFS in that it 1) implements trend tests that can be applied to failure data to determine whether it is even appropriate to apply software reliability models, and 2) the ability to combine model results according to user-specified static or dynamic weighting schemes to improve model accuracy. CASRE also has more extensive documentation than SMERFS. CASRE implements more models and has more sophisticated methods for analyzing a model's goodness of fit and applicability than do Ann Marie Neufelder's tools.

What will a successful collaboration look like?

a. **What will you as the technology provider do?** We will work with you during proposal development on planning your collaboration. For a collaboration using CASRE, we will provide a 2-day training course at your site, assistance in applying CASRE to failure data and analyzing the results, and customer telephone support over the course of the collaboration. We will also put you in touch with other groups, NASA or otherwise, using CASRE.

b. **What should the development team do?** Prior to the collaboration, the NASA software development team should be about to start or be conducting software integration or system integration testing. The testing should involve repeated sampling of the input space (which can be accomplished by regression testing after a new build). The development team should also be diligent in recording elapsed testing time (time during which the system was actually running tests) as well as failures observed during testing in order to provide accurate input to CASRE. During the collaboration, take the training course; identify the builds to which CASRE will be applied; apply CASRE, collect data on its performance, and compare with existing V&V performance data.

c. **How will you, as technology provider, work together with the development team to ensure a successful collaboration?** During the proposal process, we will start by ensuring a good technical and project match with our technology. In addition to providing customer support for CASRE, we will track the collaboration's schedule, and at appropriate points proactively contact the Collaboration PoC, learn about problems or potential problems, obtain preliminary data, offer suggestions on how to proceed, and follow up to ensure that CASRE is being used effectively and the collaboration is achieving its success criteria.